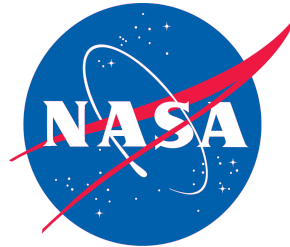
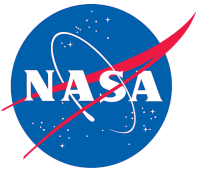

Validation of Real-Time Data Processing for the Ground and Air-MSPI Systems

Thomas Werne **Dmitriy Bekker** Paula Pingree

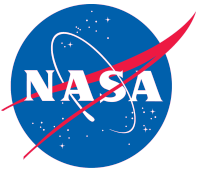
Jet Propulsion Laboratory
California Institute of Technology



March 10, 2011
IEEE Aerospace Conference
Big Sky, MT

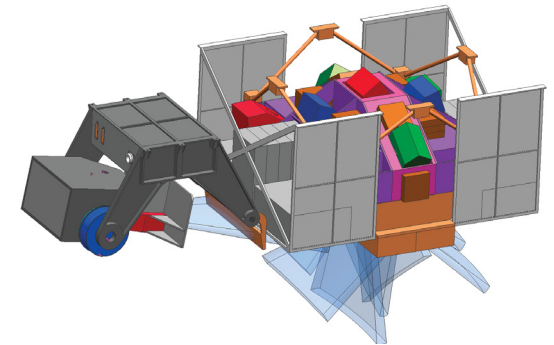


- MSPI Instrument Overview
- Onboard Processing Necessity
- Algorithm Overview and Implementation
- Phased Development Plan
- Ground-MSPI System
 - Integration
 - Validation Results
- Air-MSPI System
 - Integration
 - On-Board Processing Status
- Current Progress
- Conclusions

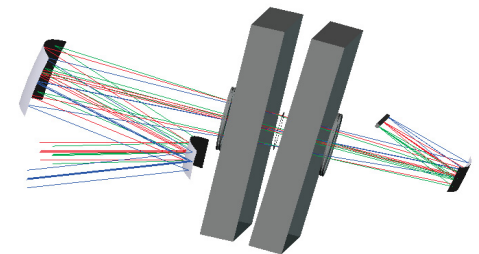


MSPI Instrument Overview

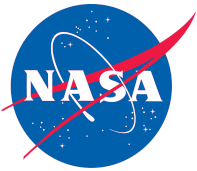
- Multiangle SpectroPolarimetric Imager (MSPI)
 - Multiangle-multiwavelength polarimeter for Aerosol-Cloud-Ecosystem (ACE) mission concept
 - Measures cloud and aerosol properties
 - 8-fixed and 1-gimballed cameras, each with 16 channels
- Design goals
 - Acquire accurate multispectral intensity imagery
 - Acquire accurate degree of linear polarization (DOLP) imagery
- Two photo-elastic modulators (PEMs) in optical path for high accuracy in DOLP
- Ground-MSPI – ground-based performance eval.
- Air-MSPI – updated design for ER-2 flight



MSPI instrument design

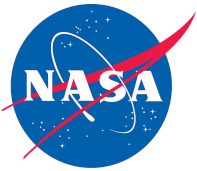


Dual PEMs in optical path



Onboard Processing Necessity

- PEMs modulate Q and U polarization components (Stokes vectors)
 - Modulated signal has frame period of 40 ms
 - Each frame is oversampled 32 times in order to create hi-fidelity digital representation of polarization components
- Each camera (one of nine), after oversampling, produces 95 Mbytes/sec of raw video data
 - Data rate too high for direct downlink from spacecraft
 - Must be processed on-board
- Using Virtex-5 FXT FPGA, we apply linear least-squares algorithm to raw data stream
 - Oversampled pixel data is reduced to five parameters per pixel in each 40 ms frame
 - No loss of information content for science – this step would be performed on the ground if no on-board processing is implemented

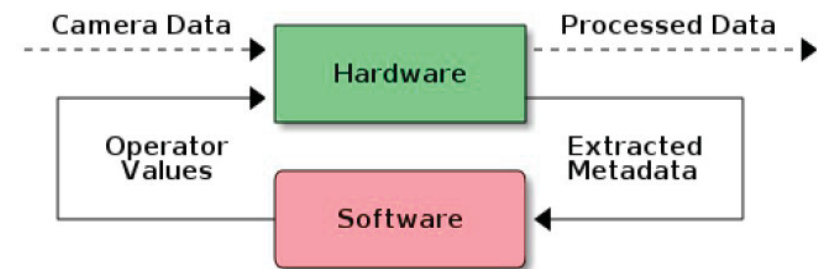


Algorithm Overview

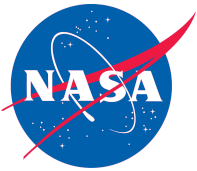
- Goal is to obtain polarization parameter estimates ($\langle I, Q, U, V \rangle$ Stokes vector) and DOLP, AOLP (functions of these parameters)
 - $\langle I, Q, U, V \rangle$ used to describe polarization state of incident light
 - I is the intensity
 - Q quantifies linear polarization at 0°
 - U quantifies linear polarization at 45°
 - V quantifies circular polarization
- MSPI on-board processing (OBP) algorithm is a self-updating multiply-and-accumulate (MAC) process
 - SW extracts metadata, computes operator values
 - HW applies operators to incoming data stream

$$DOLP = \sqrt{(Q/I)^2 + (U/I)^2} = \sqrt{q^2 + u^2}$$

$$AOLP = \frac{1}{2} \tan^{-1}(u/q)$$

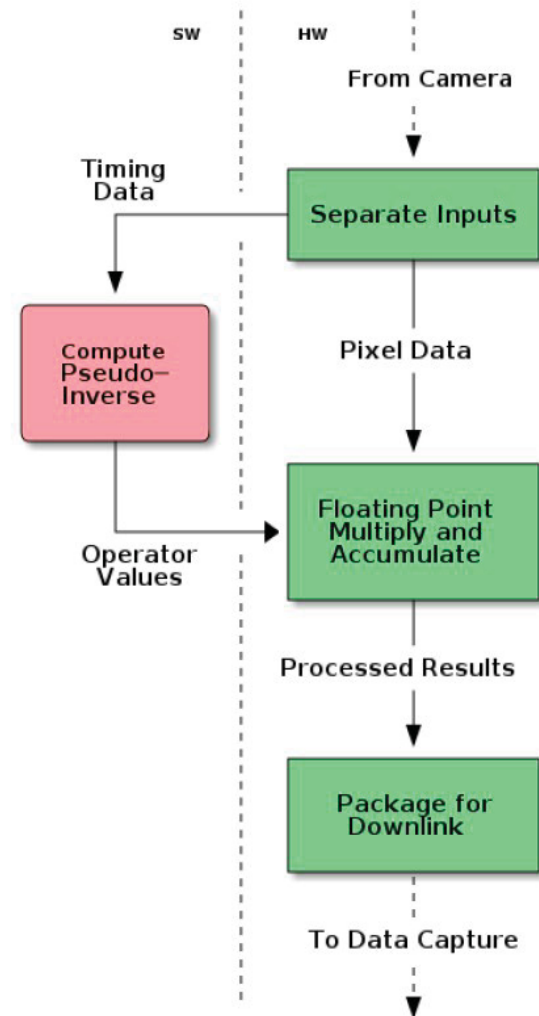


Hardware / software partitioning

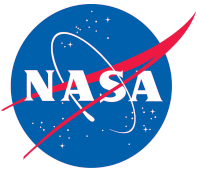


Algorithm Implementation

- In HW, separate pixel data from phase and ancillary data in stream
- In SW, use phase and ancillary data to calculate set of basis functions
- In SW, create polarization measurement matrix from sampled basis functions and calculate its pseudoinverse
- In HW, apply pseudoinverse to streaming pixel data to retrieve desired polarization parameter estimates
- Package data for output stream

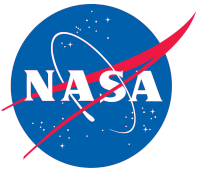


Detailed data flow



Phased Development Plan

- Project is subdivided into important technical milestones
 - Development effort proceeds in phases
 - Demonstrations occur at completion of a phase
 - Process ensures major errors are caught early
 - Leaves room for design changes based on performance of incremental builds
- Phase 1 – validate algorithm on Xilinx ML507 development board
 - Pseudo-random input data source
 - Self-contained demonstration – not connected to camera system
- Phase 2 – attach to Ground-MSPI camera data output, validate algorithm on real data, in real-time
- Phase 3 – integrate into Air-MSPI camera system, multiplex processed data into camera output stream



Ground-MSPI - Integration

- Processing of Ground-MSPI camera output is performed with Xilinx ML507 development board; 4 main system component

- CameraLink Input Interface

- Raw camera data is output to CameraLink (CL) interface
- Use external CL “breakout box” to deserialize data, feed into FPGA

- MSPI Data Processing

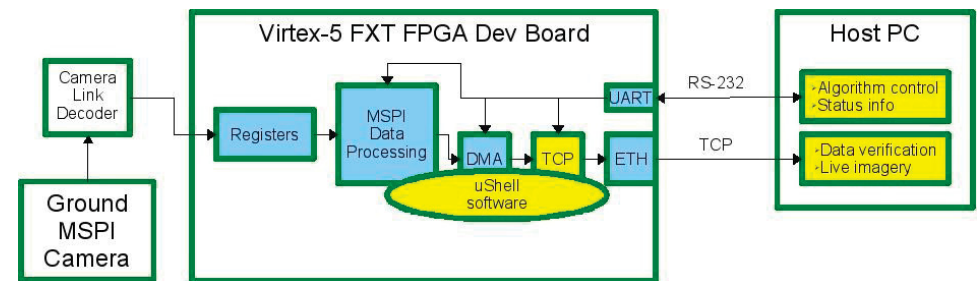
- SW computes operator values
- HW does MAC operations with dedicated DSP48 multipliers, 100 MHz

- DMA/TCP/Ethernet

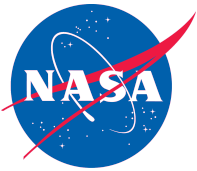
- HW Direct Memory Access transfers processed data into memory
- SW uses Lightweight TCP/IP stack to send data via Ethernet to host machine

- uShell

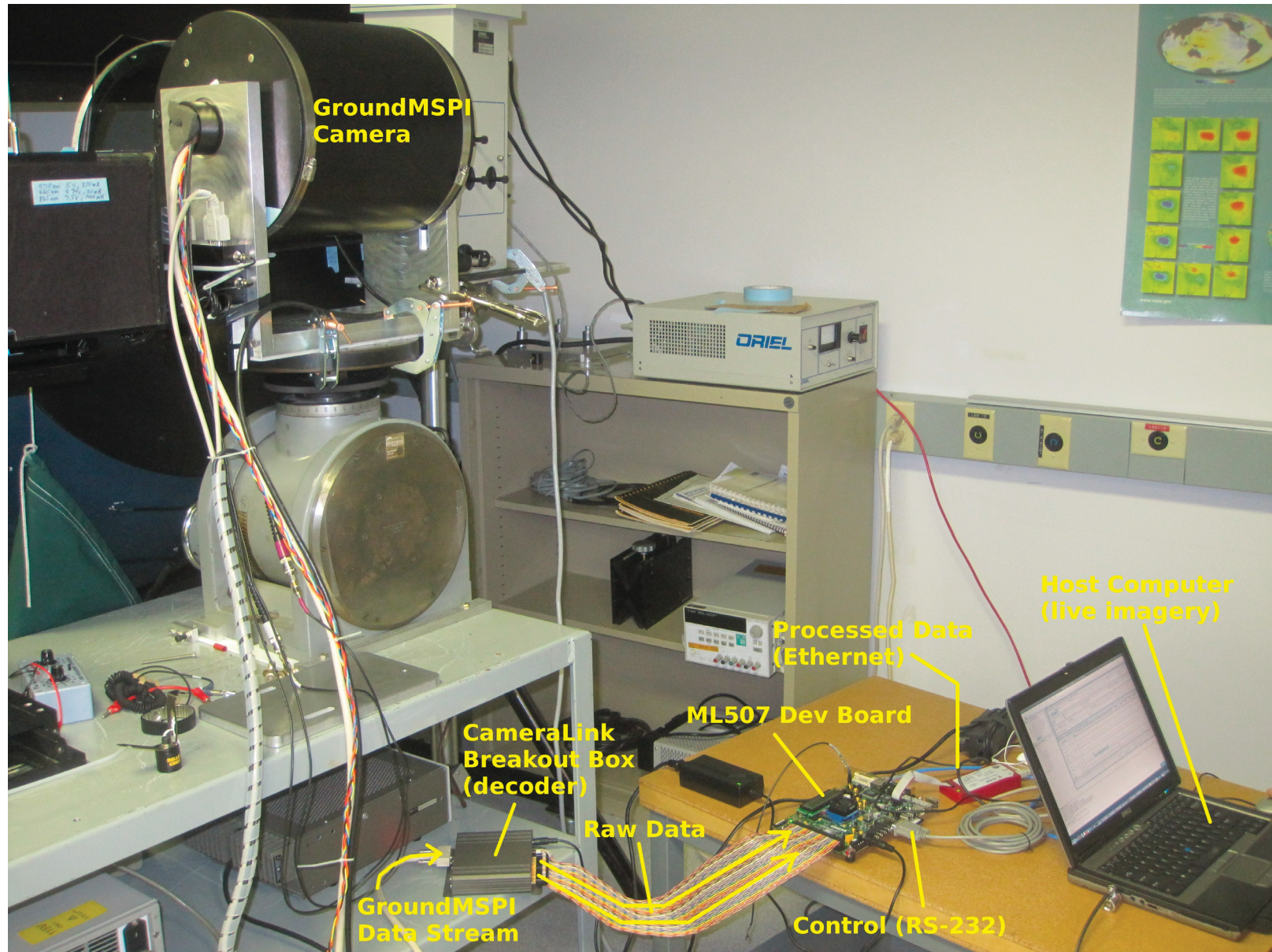
- Embedded shell environment; algorithm control, run-time debugging

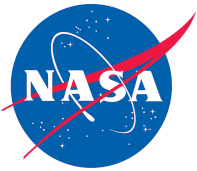


Integration scheme with Ground-MSPI board



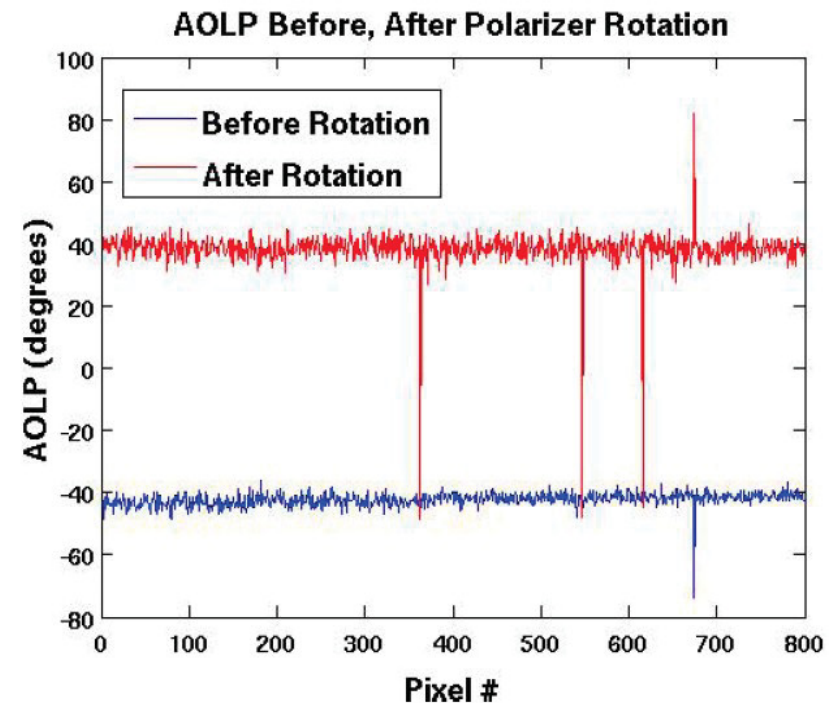
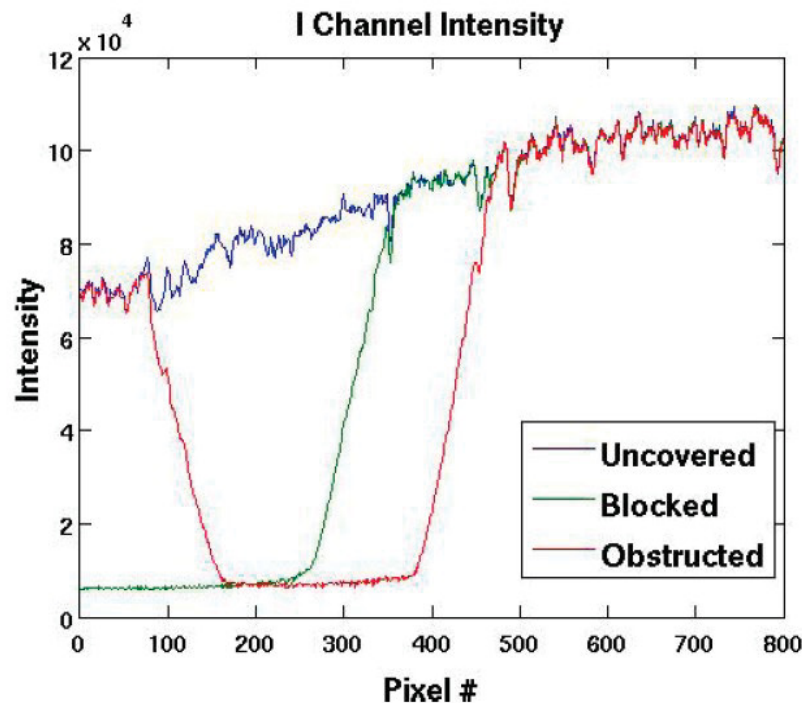
Ground-MSPI - Integration

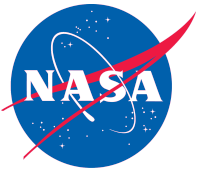




Ground-MSPI - Results

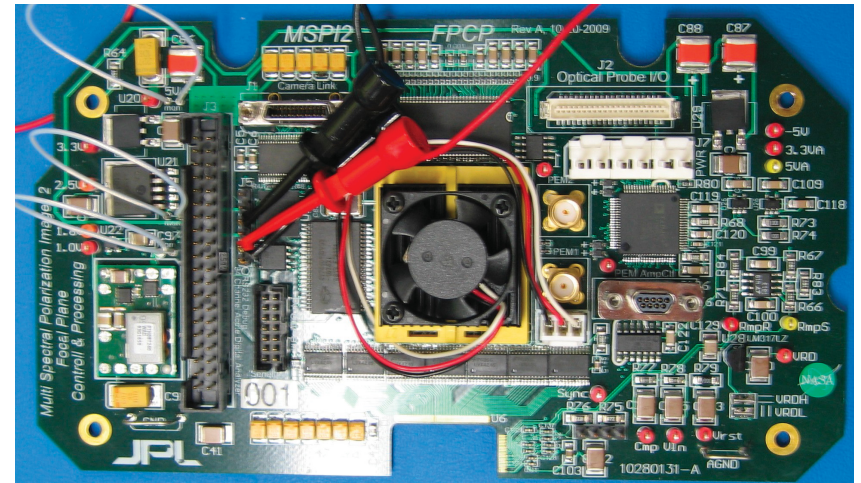
- Two experiments performed to demonstrate algorithm functionality
 - Cover portion of camera aperture, see intensity, I , decrease
 - Rotate linear polarizer in front of aperture, see AOLP change
- Spikes, non-linearities due to suboptimal camera configuration at time of experiment



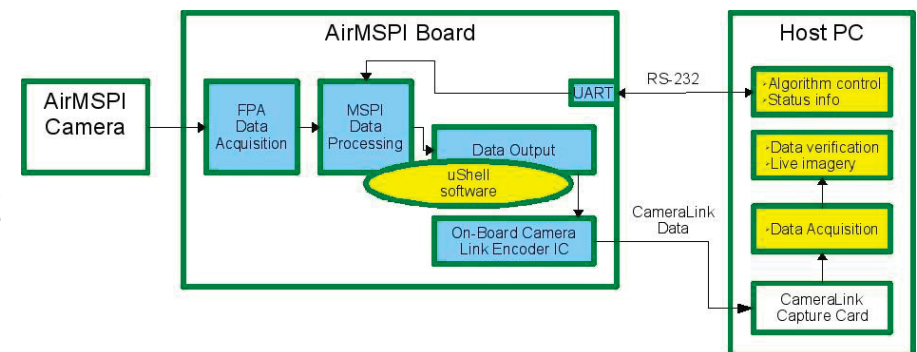


Air-MSPI - Integration

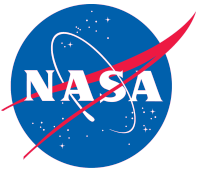
- For Ground-MSPI, we used development board for processing data on *output* of camera stream
- For Air-MSPI, we load data processing firmware *into* Air-MSPI electronics board
 - Replace CameraLink input interface with direct integration to FPA data acquisition
 - Processing unchanged
 - Replace output interface
 - No longer need DMA/TCP/Ethernet
 - Output directly to on-board CameraLink serializer
 - Multiplex processed data into raw data output stream



Air-MSPI Focal Plane Control & Processing board

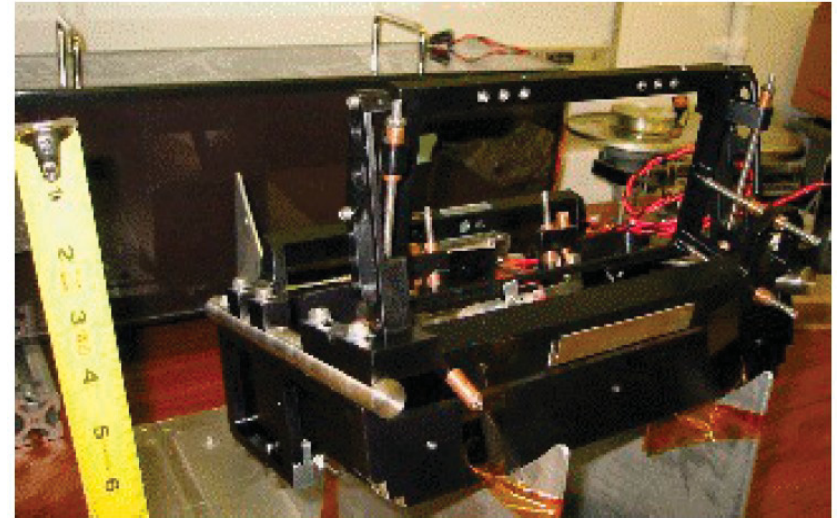


Integration scheme with Air-MSPI board



Air-MSPI – Processing Status

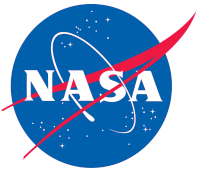
- Air-MSPI camera flew on two ER-2 flights in September 2010
 - Successfully collected raw image data
 - No on-board processing
- Currently finalizing integration of on-board processing into FPGA on Air-MSPI board
 - Very rudimentary “first-light” test successful
 - Light pixels indicate open camera aperture
 - Dark pixels indicate closed camera aperture
 - Need to update capture software to properly display floating-point data



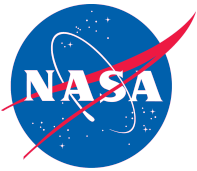
Air-MSPI Camera



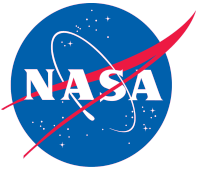
Basic “first-light” test showing light and dark bands for open and closed camera aperture (increasing time is downward)



- Design change (by MSPI instrument team) in pixel-readout timing calls for changing from Mode A to Mode C operation
 - In mode A operation (Ground-MSPI), time difference is constant between adjacent sub-frames within frame, but variable across frame boundaries
 - In mode C operation (Air-MSPI), time difference is constant both within and across frames
 - Number of sub-frames per frame is variables
 - Sub-frame timestamps no longer quasi-static
 - Basis functions need updating every frame
 - On FPGA, average sample time deltas, predict future sample times, compute set of basis functions for future sample times
- Finalizing data multiplexing of processed data into output stream
- Moving SW to soft-core MicroBlaze processor
 - For Air-MSPI, OK to use hard-core PowerPC processor on FPGA
 - For future Space-MSPI, no PowerPC available, must move to soft proc.



- Successfully demonstrated real-time processing of Ground-MSPI camera data
 - Connected to the output of camera data stream
 - Processing done on ML507 development board
- Demonstrated “first-light” of on-board data processing for Air-MSPI
 - Algorithm running on Air-MSPI board
 - Integrated with FPA data capture, CameraLink data output
- Nearly finished with remaining tasks
 - Convert processing from Mode A to Mode C
 - Multiplex processed data into raw data output stream
 - Move to a MicroBlaze implementation
- Related project “COVE” – CubeSat On-Board Processing Validation Experiment on track (scheduled launch: Oct. 2011, see conf. paper)

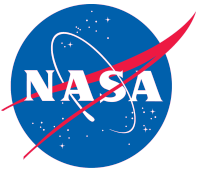


Acknowledgments

We would like to thank the MSPI science, software, and electronics teams for access to the Ground and Air-MSPI instruments, flight software, and numerous invaluable discussions and clarifications.

The research described in this presentation was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration

© 2011 California Institute of Technology. Government sponsorship acknowledged.



Backup Slide - Utilization

Feature	Occupied	Total	Utilization
Slices	4,241	11,200	37%
BRAM	92	148	62%
BUFG	10	32	31%
DSP48E	8	128	6%
PLL_ADV	1	6	16%

*FPGA utilization of MSPI OBP on Air-MSPI board
(using Virtex-5 FX70T FPGA)*